LA-UR-01-6597
Approved for public release; distribution is unlimited.

Title:

FEEMADS: Extracting Features From Hydrocode Output

Author(s): Michael Cannon, Pat Fasel, Richard Fortson and John Hogden

Submitted to:

http://lib-www.lanl.gov/la-pubs/00796524.pdf

FEEMADS:

Extracting Features From Hydrocode Output

Michael Cannon, Pat Fasel, Richard Fortson & John Hogden

CCS-3, M.S. B265

Los Alamos National Laboratory

Los Alamos NM 87545

Introduction

One of the goals of the FEEMADS project is to develop algorithms to automatically extract interesting features from hydrocode simulation outputs -- particularly RAGE hydrocode outputs (Oakes, Henning, Gittings & Weaver, 2000). As hydrocodes become more advanced, the need for such algorithms will become greater. Even now, hydrocode output is often so large that graphic displays do not have enough pixels to display the number of points in the data set. Furthermore, graphic displays are insufficient for displaying precise values of the data. For example, RAGE represents scalar values using 32 bit precision but graphic displays typically present only 8 bits of the 32 bit's dynamic range. Clearly, interpolation, scaling and changes in bit mapping can let us use graphic displays to interactively view any desired information from the hydrocode output, but for large data sets, such interactive work would be infeasible. Even if all the data could be displayed with sufficient resolution, visually tracking 3-dimensional flow data over time is a difficult problem.

It is premature to discuss extracting features without knowing something about the problem domain. Since the FEEMADS project team is composed largely of pattern recognition/machine learning specialists, but the users of the FEEMADS algorithms will be physicists, this document is intended to give a basic understanding of the goals of some of the physicists who may have use for the FEEMADS algorithms. To this end, Section 1 discusses some fundamentals of hydrodynamics codes and what is needed to determine whether the codes are working correctly. This section concludes with the observation that features that help understand turbulence and compare different turbulent fields will likely be of interest to the customer.

FEEMADS is not the only project attempting to use computers to analyze hydrocode output. Much of the field of Computer Visualization deals with displaying data similar to what is output by RAGE -- scalar or vector valued data that is organized in 2- or 3-dimensional fields (Delmarcelle & Hesselink, 1995; Hesselink & Delmarcelle, 1994; Post & Wijk, 1994). In fact, a conclusion drawn in section 1, that vortices are likely to be of interest to the FEEMADS customer, is consistent with the fact that computer visualization researchers have done extensive work to automatically detect vortices and other *critical points* in fluid flow. Some of this research, particularly work related to characterizing critical points and vortices is discussed in section 2. Since critical points are features that are clearly of interest to fluid dynamicists, some of the computer visualization work can be thought of as addressing the goals set out in FEEMADS.

Visualization of fluid flow, particularly of turbulence is difficult and is being undertaken by several laboratories. The final section lists a few laboratories doing work on data mining or visualization of hydrocode outputs.

The FEEMADS project is especially interested in analyzing RAGE hydrocode outputs. RAGE non-uniformly discretizes the volume containing the fluid. That is, the volume containing the fluid is partitioned into cubes and the fluid values (e.g., pressure, temperature, …) are constant within each cube. However, the sizes of the cubes vary – in places where the fluid properties are changing quickly with a small change in position, the cubes are smaller. The sizes of the cubes are determined by the RAGE code through a process called Adaptive Mesh Refinement (AMR).

In RAGE, the data is represented by a mesh where each grid point holds the root node of a quadtree (for two dimensional simulations) or and octree (for three-dimensional simulations). That is, each root node contains information about the partitioning of a cube into 8 sub-cubes. Each of the 8 sub-cubes holds the fluid values within that sub-cube, and can hold information about how the sub-cube is further partitioned into 8 sub-sub-cubes. The partitioning of sub-cubes can continue until the desired resolution of the spatial discretization is reached.

The octree data structure allows us to easily determine which portions of the volume are being represented with high resolution (small cubes, high mesh refinement) and which are being represented with low resolution (small cubes, low mesh refinement). Predictably, regions where the fluid properties change quickly tend to be more interesting. In fact, Norman et al. (1999, pages 44-45) observe that “the very distribution of the AMR (adaptive mesh refinement) grids indicates active areas in the computation. In a cosmological simulation, viewing animated grid-bounding boxes alone can reveal locations of filaments and forming galaxy clusters.”

The FEEMADS team has been exploring techniques for automatic feature extraction that takes advantage of the octree data structure. So far, we have found no other project that uses the level of mesh refinement as the basis for automatic feature extraction.

1. What is an interesting feature?

The extent to which a feature is interesting depends critically on the task to be performed. Since many of the potential customers for the FEEMADS projects are physicists attempting to determine whether their hydrocodes are performing correctly, features that can aid in that determination are likely to be interesting.

Thus, Section 1.1 attempts to define some of the basic terms (Lagrangian, Eulerian, Navier-Stokes, Reynolds transport theorem, …) that are frequently encountered in discussion of fluid dynamics. References to several sources that describe the relevant physics and mathematics are given for those who want a more thorough understanding.

In addition, a discussion of verification and validation, and the impact of turbulence on these processes will be discussed in section 1.2.

1.1 Fluid dynamics

The general laws of fluid dynamics are written as conservation laws, e.g., mass, energy, and momentum are conserved. Equations expressing these conservation laws are formulated by first subdividing the region containing the fluid into small, non-overlapping regions. For example, we can imagine the volume containing the fluid being subdivided into small voxels (box-shaped volume elements). Note that voxels can take on arbitrary shapes, but that box-shaped voxels are discussed here for concreteness.

The derivation then proceeds following one of two approaches. The first approach, called Lagrangian despite the fact that it was developed by Euler, is to let the voxels move with the fluid, so that each voxel always contain the same particles of fluid. In the Lagrangian approach, the position and shape of a voxel is a function of time and of the original position and shape of the voxel. For example, one text that briefly discusses the Lagrangian approach (Symon, 1971) derives equations giving the derivatives of the x,y,z position of a particular voxel (the voxel that starts at position [x0, y0, z0]) with respect to time, and the derivatives of the height, width, and length of the voxel with respect to time.

As already mentioned, the goal of this section is not to give an introduction to fluid dynamics, but to give a functional definition of common fluid dynamics terms. Nonetheless, it is instructive to see how one conservation law (we will use the conservation of mass) changes depending on the formulation. The law of conservation of mass is easy to write for Lagrangian voxels. Letting $\rho(\mathbf{x},t)$ denote the density of the fluid at time *t* at the position designated by the vector $\mathbf{x}$, the law of conservation of mass is:

$$\frac{D}{Dt}\int_v \rho(\mathbf{x},t)dV = 0 \qquad \textbf{\textit{EQ. 1}}$$

The integral in EQ. 1 gives the mass within the voxel (the V and dV indicate that the integral is taken over a volume). So the equation merely states that the mass within a voxel doesn't change over time.

Note that, instead of referring to a Lagrangian formulation, some authors write that a Lagrangian coordinate system is being used. The independent variables (the coordinates) in the Lagrangian coordinate system are time, the initial position of the voxel, and the initial shape of the voxel. Bilbao (1990) gives a nice summary of Lagrangian fluid dynamics codes:

"In all these codes velocities are assigned to the cell vertices while pressure, density, etc. are centered. When a surface value is needed, an appropriate mean value of two or more adjacent cells is calculated." The cell vertices for cubic voxels are the 8 corners of the cubes and the vertices are the 4 corners of the squares for square voxels.

Since the cell (voxel) vertices can move at different velocities, it is clear that a voxel's shape may change over time. In fact, a disadvantage of the Lagrangian coordinate system is that the voxels can be deformed by the fluid motion to such an extent that the simulation programs fail. Even without understanding the details, it is easy to imagine that the voxel deformations that could arise from vorticity could be difficult to represent. Thus a second approach to fluid dynamics is sometimes used.

The second route to formulating the conservation equations is the Eulerian approach. In contrast to the Lagrangian coordinate system, in the Eulerian approach (coordinate system), the independent variables are time and position of the voxel. In the Eulerian formulation the voxels don't move over time, so we need to characterize the fluid that is flowing into and out of the voxels. The Eulerian approach is used by the RAGE program (which is the program of interest for the FEEMEADS project), and so will be discussed in more detail than the Lagrangian approach.

The conservation laws formulated in the Eulerian coordinate system look somewhat more complex than the Lagrangian formulation. Returning to our example, the law of conservation of mass can be formulated for the motionless Eulerian voxels (as opposed to the Lagrangian moving voxels) by requiring that the change in the amount of mass in a voxel is equal to the integral of the fluid flux through the walls of the voxel. The fluid flux is the amount of fluid flowing into or out of the voxel at any point on the surface of the voxel. Thus, writing the conservation of mass equation in the Eulerian coordinate system requires some knowledge of integral vector calculus, particularly Gauss' theorem (Marsden & Tromba, 1981 is a good introduction to vector calculus). In fact, Gauss' theorem is used to derive a frequently encountered theorem in fluid dynamics, namely, Reynolds transport theorem (see Currie, 1974 for a derivation based on Gauss' theorem):

$$\frac{D}{Dt}\int_V \alpha dV = \int_V \left[\frac{\partial \alpha}{\partial t} + \nabla \bullet (\alpha \upsilon)\right] \qquad \textbf{\textit{EQ. 2}}$$

where υ is the velocity of the fluid and α is some scalar value measured at each position.

Reynolds transport theorem allows us to transform equations in the Lagrangian coordinate system into equations in the Eulerian coordinate system. Note that, if we replace α with $\rho(\mathbf{x},t)$we see that the left-hand side of Reynolds transport theorem becomes the Lagrangian expression for the change in the amount of mass inside a moving voxel (which should be 0 if mass is conserved). While it is by no means obvious, if we replace α with $\rho(\mathbf{x},t)$ and set the right-hand side of Eq. 2 to zero, we get the Eulerian formulation of the conservation of mass:

$$\int_V \left[\frac{\partial \rho(\mathbf{x},t)}{\partial t} + \nabla \bullet \left(\rho(\mathbf{x},t)\upsilon\right)\right] = 0 \qquad \textbf{\textit{EQ. 3}}$$

Eq. 3 serves only to give an example of how Reynolds transport theorem helps convert Lagrangian equations into Eulerian equations. If a more thorough understanding of the Eulerian equations for conservation of mass is required, the reader is referred to LaVeque (1992), who gives an intuitively appealing derivation of the Eulerian equations in one dimension. In one dimension, the voxels reduce to line segments and the walls of the voxels become points between the line segments. This allows the equations for the mass within a line segment to be a single integral over a line segment (as opposed to the multiple integrals used to calculate mass within a volume), and the flux is a simple expression of the mass flowing through the end-points of the line segment. For development of the equations in more dimensions, refer to either the aforementioned texts on vector calculus and fluid dynamics or, for an explanation of the differential inside the integral, to Symon (1971).

A potential source of confusion is the difference between Euler's equations and Eulerian equations. Eulerian equations, as described above, and in contrast to Lagrangian equations, are the fluid equations that have time and position as independent variables. Euler also developed a specific set of equations that describe an ideal fluid. Ideal fluids are incompressible and are non-viscous, and are considerably simpler to analyze than non-ideal fluids. They are akin to frictionless pulleys and massless ropes. In fact, a notable deficiency in the Euler's equations for an ideal gas is that they assume that the fluids are non-viscous, i.e., that a moving layer of fluid will not affect the velocity of adjacent layers of fluid. The Navier-Stokes equations describe fluids that are viscous and/or compressible.

Note that the equation for the conservation of mass is written as an integral over a volume. In the integral form, the equation can be applied even when one of the derivatives inside the integral is discontinuous. For example, if the equations were being applied to a tube having a shock wave travelling through it, then velocity could be discontinuous so the gradient of the velocity may not exist for some locations. In the integral form of the equation, however, the integral could still be calculated – it would only require doing the integral separately on the sides of the discontinuity and then summing the integrals.

The equation for the conservation of mass (and the other fluid dynamics equations) can also be written in a differential form:

$$\frac{\partial \rho(\mathbf{x},t)}{\partial t} + \nabla \bullet \left(\rho(\mathbf{x},t)\upsilon\right) = 0 \qquad \textbf{\textit{EQ. 4}}$$

The justification for this formulation is that the differential must be zero for all locations at which it exists in order for the integral equation to be zero. However, in order for the differential form to be accurate, the derivatives must exist at all locations, and so it does not apply to cases where a discontinuity (e.g. a shock wave) exists. Discontinuities are not just a theoretical problem, they often result in numerical difficulties in hydrocodes, which are based on the differential forms of the conservation laws.

There are ways to get around the discontinuity problem. For example, since the differential forms are valid in regions that don't have discontinuities, we can track the discontinuity over time and use the differential forms in areas not containing the discontinuities. This is a motivation for some of the work on tracking shock waves (e.g. Sethian, 1999).

One other approach (but not the only other approach) is to add viscosity to the equations. Using the Navier-Stokes equations instead of Euler's equations can alleviate problems due to discontinuities because discontinuities are smoothed out over time in a viscous fluid. Note that the authors are not sure whether using the Navier-Stokes equations are sufficient to eliminate discontinuity problems. Our concern is that, while Navier-Stokes equations may smooth out discontinuities, the discretization may be sufficiently coarse that the discontinuities still appear to be present.

1.2 Verification and Validation

There are a variety of problems for which the fluid dynamics equations are too difficult to solve explicitly. RAGE and other hydrocodes provide numerical approximations to the equations, allowing us to estimate fluid flow in new situations. To ensure that the programs give good estimates of what real fluids will do, we need to know that the equations that the programs are solving are the appropriate equations for the problem, and that the equations are being solved correctly. *Validation* is a term used to describe the process of determining whether we are solving the correct equations – the equations that accurately describe the fluid. *Verification* is the term used to describe whether the program solves the equations correctly, i.e., does the program have bugs, is the step size in the numerical integration small enough, etc. These terms are becoming standard in the literature (AIAA, 1998).

It is easy to confuse validation and verification, but a relatively simple mnemonic can help remember which is which. Letters 2-4 of validaton (“a”, “l”, “i”, “d”) can be taken to be the first letters of the phrase “accurate laws in description”, as in the conservation laws used correctly describe the problem. On the other hand, the letters “e”, “r”, “f”, and “c” from verification can be taken as the first letters of the phrase “errors removed from code”.

Any feature that would help detect a problem with the validity of the equations or detect a flaw in the program implementing the equations is clearly an interesting feature. Since such features are important, we will discuss some work that has been done to verify and validate the programs. While verification is certainly problematic for complex problems (Gustafson, 1998), more emphasis will be placed on validation, since this is the realm that seems more likely to suggest which features should be extracted.

1.2.1 Verification

Verification of hydrodynamic simulations can be done without experimental data. One way to do so is to find problems that can be solved analytically and compare the program output to the known correct solution. A variety of fluid dynamics problems can be solved analytically (Coggeshall, 1991 gives some examples and references to many more; Reinicke & Meyer-ter-Vehn, 1991). Some of these problems have also been used for verification of RAGE (and other) hydrocodes (Clover, Kamm & Rider, 2000; Kamm et al., 1999). The RAGE code performed fairly well on these (admittedly not exhaustive) tests.

Comparisons between hydrocode output and the known analytic solutions are relatively simple, e.g., the sum squared differences between the fluid values for the simulation and the exact solution can be found. However, the fluid flow problems that have analytic solutions tend to be much simpler than the problems of interest to FEEMADS. Validating the results of a fluid flow simulation is made very difficult by the fact that most fluid flows result in some amount of turbulence. As discussed in the next section, turbulence complicates the validation problem tremendously.

1.2.2 Importance of Turbulence in Validation

Many simple initial fluid configurations lead to complex solutions containing turbulence, for which a slight imprecision in the initial fluid flow description can lead to simulation results that are correct in some sense (e.g., vortices are created) but for which experimental measurements and the simulation outputs can be very different on a voxel-by-voxel basis. Thus, comparing simulated turbulent fluid flow to experimental measurements requires calculating large scale statistical features of the flow. The ubiquity of turbulence is discussed in section 1.2.2.1 and potential features (specifically, vorticity) are discussed in section 1.2.2.2.

1.2.2.1 An example Turbulent Validation Problem: The Rayleigh-Taylor Instability

Fluid dynamics problems involving interfaces between two or more fluids commonly lead to turbulent solutions. Consider the Rayleigh-Taylor instability – which is the result of a very simple initial configuration of two fluids. An overview of the Rayleigh-Taylor instability by Sharp (1984) asks us to “imagine the ceiling of a room plastered uniformly with water to a depth of 1 meter”. It is then pointed out that the pressure of the atmosphere is sufficient to support the water. Nonetheless, we know that the water will fall. The water falls because the interface between the water and the air is not perfectly planar (this has to be the case because the fluids are not actually continuous media). Any infinitesimal deviation from a planar interface will grow over time, until the water is on the floor.

If we were to simulate the room with water plastered to the ceiling, the simulation would give the correct solutions in some ways. For example, pockets of air would propagate upward, vortices would likely be created, the water would end up on the floor, etc. However, unless we had the exact description of the interface between water and air, details such as where the water comes down and where the air goes up could not possibly be determined. If we were only interested in the steady state solution to the problem (where the water is on the floor) then comparisons between a simulation and the known solution would, again, be simple. However, simulating the steady state solution is not sufficient -- the dynamics of fluid mixing are critical for many fields of study. In situations of turbulent dynamics, voxel-by-voxel comparisons of fluid values are of little use.

Approximate solutions of the fluid dynamics for the Rayleigh-Taylor instability for very short time intervals exist, and the flow can be characterized to some extent over longer intervals, but an analytic solution to this problem has not been found (Currie, 1974; Sharp, 1984; Youngs, 1984). Since we cannot find an analytic solution for an exactly specified fluid interface, hydrocodes cannot be directly verified (we can't know if the code is a correct implementation of the mathematics) in the case of the Rayleigh-Taylor instability. Nonetheless, if the simulation gave the exactly correct fluid dynamics compared to an experiment, we could consider the code to be verified and the conceptual model to be validated (at least for that one situation). Unfortunately, we cannot compare the simulation to an experiment on a voxel-by-voxel basis either, because we can't know the exact shape of the fluid interface.

Of course, turbulence can be generated in many other ways as well. In general, if the interface between two fluids of different density is uniformly accelerated toward the less dense fluid, the Rayleigh-Taylor instability will be observed. In fact, in some experimental studies of the Rayleigh-Taylor instability a container is partially filled with fluid and accelerated downwards, so that the interface is accelerated toward the less dense fluid, air (Read, 1984). A fluid interface accelerated by a shock also generates turbulence – in this case called the Richtmyer-Meshkov instability (Richtmyer, 1960). Shocks interacting with turbulence amplify the turbulence (Andreopoulos, Agui & Biassulis, 2000). Turbulence will also be observed in the flow of a fluid through a pipe, and is, in fact, common in real-world problems.

According to Shivamoggi (1998, chapter 6), one characteristic of turbulence is "increased diffusivity which causes rapid mixing and enhanced rates of momentum, heat and mass transport." The ability of turbulence to accelerate fluid mixing suggests a reason why turbulence is important: in order to understand fluid mixing (which affects combustion rates and, clearly, the purity of materials) we need to be able to understand turbulence. Furthermore, the enhanced rates of momentum transfer affect predictions about the energy that a shock wave can impart on a turbulent field, and therefore the amount of energy left in a shock wave that passes through a turbulent field (Andreopoulos et al., 2000). Combining these reasons with the ubiquity of turbulence in real-world fluid flow simulations and the need to validate code gives ample justification for an attempt to better understand turbulence.

1.2.2.2 Features of Turbulence

Big whorls have little whorls
Which feed on their velocity,
And little whorls have lesser whorls
And so on to viscosity – Lewis Richardson

Since it is still an open question as to which measures or features of turbulence are most important, it would undoubtedly be worthwhile for the FEEMADS team to understand turbulence in more depth. Unfortunately, a thorough discussion of turbulence cannot be provided here. However, in the interest of inspiring ideas about finding features that characterize turbulence, a brief discussion of features that may be of interest in characterizing turbulence is given. As a result of the discussion, our attention will focus on vorticity as an interesting feature of turbulence, while not ruling out other avenues.

Fourier analysis, fractal dimension, wavelet analysis, and other techniques have been used to analyze the similarity between simulation and experiment in turbulent data flow (Kamm et al., 1999; Rider & Kamm, 2000). A good overview of turbulence, and the use of fractals, wavelets, and Fourier transforms to characterize it is given by Debnath (1998).

There are also various mathematical descriptions of turbulence (Shivamoggi, 1998, chapter 6) that may suggest which features should be extracted to compare turbulence in a simulation to turbulence observed in an experiment. For example, some theoretical developments describe the way energy will be transferred from large-scale structure (e.g., large vortices) to smaller-scale structures (e.g., smaller vortices). This theory suggests that there should be a characteristic region of the Fourier transform of turbulence which is relatively stable over time (Shivamoggi, 1998, pages) and which is not strongly affected by the large scale structure of the turbulence.

While turbulence is characterized by unstable fluid flows, it is not devoid of structure. In fact, Debnath (1998, pages 137-141) mentions (and gives references to) a variety of coherent structures that have been observed in turbulent flow. In the words of Debnath: "Coherent structures are organized spatial features which repeatedly occur and undergo a characteristic temporal life cycle." Any of these coherent structures could provide the basis for a feature to be detected.

One fundamental coherent structure in turbulence is the vortex. In fact, a large body of research focussed on how vortices can help us understand turbulence has been reviewed by Zabusky (1999). Zabusky's review is likely to be of particular interest to the FEEMADS team because it lists a variety of coherent structures that are candidate features to extract from hydrodynamics code outputs. Zabusky also briefly mentions some relevant feature extraction work, including some of the techniques reviewed in Section 2, below.

2. Current Feature Extraction Research

As previously mentioned, much of the work in visualization can be thought of as extracting interesting features (streamlines, isosurfaces, critical points, …) from hydrodynamic simulations. However, the visualization programs are typically more interactive than FEEMADS desires. For example, to see a streamline, the user may be prompted to specify a point in the simulation and then the program shows where that point travels over time. However, specifying the points to get a useful image can be difficult. In fact, Roth & Peikert (1998) write: "Classical visualization techniques require careful and laborious selection of streamlines or isosurface level to display the essential properties of a flow." Some newer, automatic techniques for generating streamlines may decrease the difficulty (Mao, Hatanaka, Higashida & Imamiya, 1998).

Work has also been done to track features (e.g., vortices) over time (Villasenor & Vincent, 1992). The feature tracking work done at Rutgers University (Bitz & Zabusky, 1990; Samtaney, Silver, Zabusky & Cao, 1994; Silver & Kusurkar, 2000; Silver & Wang, 1997; Silver & Wang, 1998; Zabusky, 1999) is particularly relevant to FEEMADS. The Rutgers group has done extensive work on tracking features in hydrodynamics simulations – including detecting important events like the creation of an amorphous object, the disappearance of an amorphous object, and the merging of two objects. They have developed software for distributed processing to speed up the feature tracking. The tracking program developed by the Rutgers group works on scalar fields. For example, the Rutgers technique has been studied for its use in extracting vorticity (the magnitude of the curl of a vector field) which is related to vortices, as we discuss in Section 2.2.

While vortices are particularly relevant in analyzing turbulence, vorticity may not be the best indicator of a vortex. In the next section we describe some of the other techniques use to find vortices.

2.2 Theoretical background of vortex detection

Vorticity, the magnitude of the curl of a vector field, is not a sure indicator of a vortex. One of the simplest examples of an instance where the vorticity is non-zero but there is no vortex is a shear flow. For example, suppose we have a 3-dimensional simulation in which the velocity of the fluid in the direction parallel to the y axis is always 0, the velocity in the direction of the z axis is zero, but the velocity in the x direction is y^2. Clearly, in this case, there is no vortex and yet the curl of the vector field is always 2y in the direction of the z axis, which is non-zero except on the y=0 plane. Shear flow can present real problems for detecting vortices using vorticity, as noted by Jeong et al. (1997) who wrote "To educe CS (coherent structures) and study their dynamics, we must first develop a means to extract vortex cores directly from the instantaneous velocity field. For this purpose, the vorticity magnitude is a poor choice, since high vorticity magnitude is present everywhere near the wall, due to background shear." So while vorticity has been successfully used by the Rutgers group to better understand vortices, further analysis can be used to better detect vortices.

In order to better understand the work that has been done to detect vortices, it is important to understand what a critical point is, and the importance of a critical point in visualizing fluid flow. While critical points are a standard topic in differential equations (Boyce & DiPrima, 1977), the current prominence of critical points in visualization can be credited to Helman and Hesselink (Delmarcelle & Hesselink, 1995; Helman & Hesselink, 1989; Helman & Hesselink, 1991; Hesselink & Delmarcelle, 1994).

A critical point is a position in a vector field where the vector magnitude is 0. If the vector field is a velocity field, then the velocity is 0. Consider a single time slice of a fluid flowing in 2-dimensions. Assume we know the velocity of the fluid at each point in the plane and we only consider velocity as a function of position, not time. Let us denote the velocity vector:

$$\mathbf{V}(x,y) = F(x,y)\hat{i} + G(x,y)\hat{j} \qquad \text{EQ. 5}$$

where G and H are scalar functions and $\hat{i}$ and $\hat{j}$ are unit vectors parallel to the x and y axes respectively. A critical point is a point for which

$$dx/dt = F(x,y) = 0 \quad \& \quad dy/dt = G(x,y) = 0. \qquad \text{EQ.6}$$

Critical points can help a researcher find good places to start calculating streamlines, because streamlines started a short distance from a critical point are particularly informative. Normally, a streamline can be calculated solving:

$$dy/dx = \frac{dy/dt}{dx/dt} = G(x,y)/F(x,y) \qquad \text{EQ.7}$$

Neither this, nor the reciprocal can be calculated for critical points because it would require dividing by 0. In other words, the slope of the streamline is indeterminate at critical point but not indeterminate elsewhere. This kind of indeterminacy is caused by streamlines that converge to the same point but which do not have the same tangent at the point of convergence. For example, if two flow lines cross, then the slope of the flow line at the crossing point is indeterminate – the slope could be the slope of either of the flow lines. This indeterminacy occurs at the center of a vortex, since, as we approach the center of a vortex from the 0 degree direction the streamlines are perpendicular to the streamlines we would observe if we approach the center of the vortex from the 90 degree direction. Therefore, vortex centers are critical points.

While all vortex centers in two dimensions are critical points, not all critical points are vortex centers. Since there is no flow at a critical point, critical points can be classified into various types depending on what the streamlines look like in a region close to the critical point. Figure 1 shows a simple classification of the basic types of critical points:

1) Attracting/repelling star: stream lines go toward/away from the critical point.
2) Attracting/repelling focus: stream lines swirl into/away from the critical point in a vortex
3) Center: stream lines travel around the critical point without getting closer or farther away
4) Saddle point: streamlines go toward the critical point from some directions and away from the critical point from other directions.

The type of critical point can be determined from the Jacobian of the vector field at the critical point. Note that the velocity at some small distance from the critical point can be approximated by:

$$\begin{aligned} F(x+\Delta x, y+\Delta y) &\cong \frac{\partial F}{\partial x}\Delta x + \frac{\partial F}{\partial y}\Delta y + F(x,y) \\ G(x+\Delta x, y+\Delta y) &\cong \frac{\partial G}{\partial x}\Delta x + \frac{\partial G}{\partial y}\Delta y + G(x,y) \end{aligned} \qquad \text{EQ. 8}$$

When (x,y) is the location of a critical point, the F and G terms on the right hand side are zero, so the equations can be written in matrix format as:

$$\begin{bmatrix} F(x+\Delta x, y+\Delta y) \\ G(x+\Delta x, y+\Delta y) \end{bmatrix} = J \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \qquad \text{EQ. 9}$$

with J being the Jacobian:

$$J = \begin{bmatrix} \partial v_x / \partial x & \partial v_x / \partial y \\ \partial v_y / \partial x & \partial v_y / \partial y \end{bmatrix} \qquad \text{EQ. 10}$$

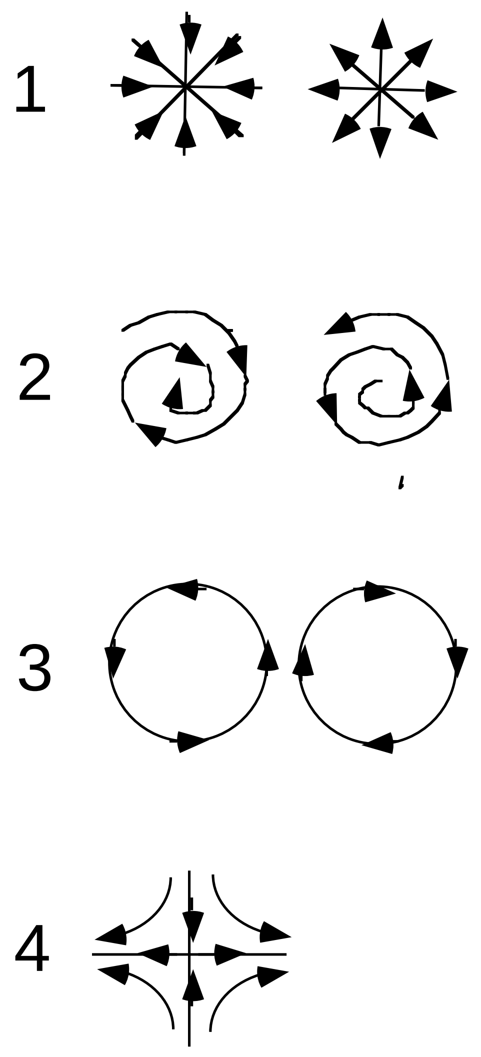

Figure 1

If the eigenvalues of the Jacobian matrix are real, the node is either a repelling star, attracting star, or saddle point. An easy way to see this is to think about what happens to the velocity as we move away from the critical point in the direction of an eigenvector of the Jacobian. Suppose that $\hat{i}$ (the unit vector parallel to the x-axis) is an eigenvector of the Jacobian with a positive, real eigenvalue, λ. If we move away from the critical point in the direction of $\hat{i}$ then the velocity will be $\lambda\hat{i}$. That is, the velocity will be pointing away from the critical point in the direction $\hat{i}$. If λ is real and negative, then the resulting velocity vector will point toward the critical point. Thus, if both eigenvalues are real and positive the critical point is repelling, if both eigenvalues are real and negative the critical point is attracting, and a saddle point results if both eigenvalues are real but one is negative and the other is positive.

If the eigenvalues are complex, they will occur as a conjugate pair. In particular, if one eigenvalue is the imaginary number *i* (not to be confused with the unit vector) then the other eigenvalue must be *–i*. To see the effect of a purely imaginary eigenvalue, it is helpful to think in terms of the symmetric and antisymmetric parts of the Jacobian. Any matrix can be broken into symmetric and antisymmetric components

$$J = \frac{1}{2}(J + J^T) + \frac{1}{2}(J - J^T) \qquad \text{EQ.11}$$

Where the symmetric component is $J_s = \frac{1}{2}(J + J^T)$ EQ. 12

And the antisymmetric component is $J_a = \frac{1}{2}(J - J^T)$ EQ. 13

If the Jacobian is symmetric, it will have real eigenvalues (Strang, 1980) and if the Jacobian is antisymmetric, it will have purely imaginary eigenvalues. Notice that, in 2 dimensions, an antisymmetric matrix Jacobian is particularly simple – it always takes the form:

$$\begin{bmatrix} 0 & a \\ -a & 0 \end{bmatrix} \qquad \text{EQ. 14}$$

Multiplying any vector by an antisymmetric matrix with a>0 always returns a vector rotated 90 degrees clockwise compared to the original vector. If a<0 the resulting vector is rotated 90 degrees counterclockwise from the original vector. This can be proven by multiplying by an antisymmetric matrix by a unit vector of arbitrary direction and noticing that the resultant is just a rotated and scaled version of the original:

$$\begin{bmatrix} 0 & a \\ -a & 0 \end{bmatrix}\begin{bmatrix} \sin\theta \\ \cos\theta \end{bmatrix} = \begin{bmatrix} a\cos\theta \\ -a\sin\theta \end{bmatrix} = a\begin{bmatrix} \cos\theta \\ -\sin\theta \end{bmatrix} \qquad \text{EQ. 15}$$

Thus, when the eigenvalues of the Jacobian evaluated at the critical point are purely imaginary, the velocity a short distance away from the critical point is perpendicular to the direction moved, and the critical point is a vortex center. When the eigenvectors are complex, the Jacobian has a symmetric part and an antisymmetric part so the result is a linear sum of vortical motion and some amount of repulsion or attraction, i.e., the repelling focus or an attracting focus or (an option that isn't shown in figure 1) a focus that is repelling in some regions and attracting in some other regions. The last option is found when the symmetric part of the Jacobian has one negative and one positive eigenvalue and there is an antisymmetric part of the Jacobian.

The preceding discussion is limited to two dimensions, but similar reasoning can be used to extend the classification of critical points to three dimensions (Chong, Perry & Cantwell, 1990). Of course, the critical point concept can be extended in many other directions as well. For example, by using the Jacobian we are approximating a complex function using only linear terms. While this is certainly valid over a small enough region, when using simulation output there is a lower limit to the size of the region we may consider (space is discretized in hydrocode outputs) and the linear approximations may not be sufficiently accurate. So it may be necessary to use higher-order terms, which lead to more complex categories of critical point and more complex interpolation methods in analyzing critical points (Scheuerman, Hagen, Kruger, Menzel & Rockwood, 1997).

The critical point techniques are also focussed on finding the core of the vortex, not the whole vortex, and so can be supplemented (or replaced by) other techniques. In fact, Banks & Singer (1994) summarize eight techniques for finding vortices, including a technique based on analyzing the Jacobian, before describing their own technique, based on combining a vorticity measurement with a search for a local pressure minimum. It is also possible to combine critical points to simplify a display (Leeuw & Liere, 1999). Undoubtedly, many more techniques and techniques based on combining techniques will be developed.

3. Other FEEMADS related work

In the course of doing this literature review, we found several sites that appear to be doing work generally related to FEEMADS – work involving data mining on hydrocode output. These sites are listed here for future reference.

Lawrence Livermore's Center for Applied Scientific Computing is developing Sapphire, "an object oriented framework for the interactive exploration of large, complex, multidimensional scientific data" (CASC,). This framework is intended to allow users to try a variety of pattern recognition and data mining algorithms.

Argonne National Laboratory is also developing algorithms for allowing interactive visualization of very large data sets and has implemented various parallel algorithms intended to extract a portion of a data set for visualization (Freitag & Loy, 2000 also see web site). For example, a uniformly sampled grid can be extracted from octree data.

Lawrence Berkeley National Laboratory's National Energy Research Scientific Computing Center (NERSC) is developing Adaptive Mesh Refinement computing tools, including a visualization toolkit (Ligocki, 2000).

The ASCI program has also created five University-based centers of excellence for studying computational problems related to Defense Programs. These centers do simulation and data mining work which is relevant to the FEEMADS program. Perhaps the most closely related is Stanford's Center for Integrate Turbulence Simulations (CITS) which is doing data mining work on simulation outputs. CITS is particularly interesting because Prof. Jerome Friedman (better known for his work in machine learning) is leading the data mining effort (CITS, See the web site for more info.). Other centers of excellence are: California Institute of Technology's Virtual Facility for Simulating Dynamic Response of Materials, the University of Chicago's Astrophysical Thermonuclear Flashes Center, The University of Utah's Center for Simulation of Accidental Fires and Explosions, and the University of Illinois' Center for Simulation of Advanced Rockets.

NASA is also doing turbulence research at the Center for Turbulence Research, but with little work on data mining.

References

AIAA. (1998). Guide for the Verification and Validation of Computational Fluid Dynamics Simulations (AIAA Guide G-077-1998).

Andreopoulos, Y., Agui, J. H., & Biassulis, G. (2000). Shock wave-turbulence interactions. Annual Review of Fluid Mechanics, 32, 309-345.

Banks, D., & Singer, B. (1994). Vortex tubes in turbulent flows: Identification, representations, reconstruction. IEEE Visualizations 1994 Proceedings, 132-139.

Bilbao, L. (1990). A three-dimensional Lagrangian method for fluid dynamics. Journal of Computational Physics, 91, 361-380.

Bitz, F. J., & Zabusky, N. J. (1990). David and visiometrics: visualizing and quantifying evolving amorphous objects. Computers in Physics, 4(6), 603-613.

Boyce, W. E., & DiPrima, R. C. (1977). Elementary Differential Equations. (3rd Edition ed.). New York: John Wiley & Sons.

CASC. Sapphire: Large-scale Data Mining and Pattern Recognition (UCRL-TB-132076). Livermore: Lawrence Livermore National Laboratory.

Chong, M. S., Perry, A. E., & Cantwell, B. J. (1990). A general classification of three-dimensional flow fields. Physics of Fluids A, 2(5), 765-777.

CITS. Simulation Implementation Plan. , 8.

Clover, M. R., Kamm, J. R., & Rider, W. J. (2000). An example of verification analysis for an Eulerian hydrocode (LA-UR-00-5371). Los Alamos, NM: Los Alamos National Laboratory.

Coggeshall, S. V. (1991). Analytic solutions of hydrodynamics equations. Physics of Fluids A, 3(5), 757-769.

Currie, I. G. (1974). Fundamental Mechanics of Fluids. New York: McGraw-Hill Book Company.

Debnath, L. (1998). Wavelet transforms, fractals and turbulence. In L. Debnath & D. N. Riahi (Eds.), Nonlinear Instability, Chaos and Turbulence, (Vol. Volume 1, pp. 129-195). Southhampton, UK: Computational Mechanics Publications/WIT Press.

Delmarcelle, T., & Hesselink, L. (1995). A unified framework for flow visualization. In R. S. Gallagher (Ed.), Computer Visualization, (pp. 129-170). London: CRC Press.

Freitag, L. A., & Loy, R. M. (2000). Using desktop graphics workstations for interactive remote exploration of large data sets. Visualization Development Environments 2000, 1-6.

Gustafson, J. (1998). Computational verifiability and feasibility of the ASCI program. IEEE Computational Science and Engineering, January-March, 36-45.

Helman, J., & Hesselink, L. (1989). Representation and display of vector field topology in fluid flow data sets. Computer, 22(8), 27-36.

Helman, J. L., & Hesselink, L. (1991). Visualizing vector field topology in fluid flows. IEEE Computer Graphics & Applications, 11(3), 36-46.

Hesselink, L., & Delmarcelle, T. (1994). Visualization of vector and tensor data sets. In L. Rosenblum, R. A. Earnshaw, J. Encarnacao, H. Hagen, A. Kaufman, S. Klimenko, G. Nielson, F. Post, & D. Thalman (Eds.), Scientific Visualization: Advances and Challenges, (pp. 419-433). New York: Academic Press.

Jeong, J., Hussain, F., Schoppa, W., & Kim, J. (1997). Coherent structures near the wall in a turbulent channel flow. Journal of fluid Mechanics, 332, 185-214.

Kamm, J. R., Rider, W. J., Vorobieff, P. V., Rightley, P. M., Prestridge, K. P., & Benjamin, R. F. (1999). Verification and validation of hydrocodes with detailed experimental data (LA-UR-99-5575). Los Alamos, NM: Los Alamos National Laboratory.

Leeuw, W. d., & Liere, R. v. (1999). Collapsing flow topology using area metrics. IEEE Visualization 99 Proceedings, 349-354.

LeVeque, R. J. (1992). Numerical Methods for Conservation Laws. (2nd Edition ed.). Berlin: Birkhauser Verlag.

Ligocki, T. J. (2000). Implementing a visualization tool for adaptive mesh refinement data using VTK. Visualization development Environments 2000 Proceedings, 1-9.

Mao, X., Hatanaka, Y., Higashida, H., & Imamiya, A. (1998). Image-guided streamline placement on curvilinear grid surfaces. IEEE Visualization 98 Proceedings, 135-142.

Marsden, J. E., & Tromba, A. J. (1981). Vector Calculus. (2nd edition ed.). New York: W. H. Freeman and Company.

Norman, M. L., Shale, J., Levy, S., & Daues, G. (1999). Diving deep: data-management and visualization strategies for adaptive Mesh Refinement Simulations. Computing in Science and Engineering, 1(4), 36-47.

Oakes, W. R., Henning, P. J., Gittings, M. L., & Weaver, R. P. (2000). On 3D, automated, self-contained grid generation within the RAGE CAMR hydrocode (LA-UR 00-2996). Los Alamos, NM: Los Alamos National Laboratory.

Post, F. H., & Wijk, J. J. v. (1994). Visual representation of vector fields: recent developments and research directions. In L. Rosenblum, R. A. Earnshaw, J. Encarnacao, H. Hagen, A. Kaufman, S. Klimenko, G. Nielson, F. Post, & D. Thalman (Eds.), Scientific Visualization, (pp. 367-390). New York: Academic Press.

Read, K. I. (1984). Experimental investigation of turbulent mixing by Rayleigh-Taylor instability. Physica D, 12, 45-58.

Reinicke, P., & Meyer-ter-Vehn, J. (1991). The point explosion with heat conduction. Physics of Fluids A, 3(7), 1807-1818.

Richtmyer, R. D. (1960). Taylor Instability in shock acceleration of compressible fluids. Communications in Pure and Applied Mathematics, 13, 297.

Rider, W. J., & Kamm, J. R. (2000). The impact of numerical integration on gas curtain simulations (LA-UR-00-5464). Los Alamos: Los Alamos National Laboratory.

Roth, M., & Peikert, R. (1998). A higher-order method for finding vortex core lines. IEEE Visualization 98 Proceedings, 143-150.

Samtaney, R., Silver, D., Zabusky, N., & Cao, J. (1994). Visualizing features and tracking their evolution. IEEE Computer, 27(7), 20-27.

Scheuerman, G., Hagen, H., Kruger, H., Menzel, M., & Rockwood, A. (1997). Visualization of higher order singularities in vector fields. IEEE Visualization 1997 Proceedings, 67-74.

Sharp, D. H. (1984). An overview of Rayleigh-Taylor instability. Physica D, 12, 3-18.

Shivamoggi, B. K. (1998). Theoretical Fluid Dynamics. (2nd edition ed.). New York: John Wiley & Sons, Inc.

Silver, D., & Kusurkar, Y. (2000). Visualizing time varying distributed datasets. Visualization Development Environments 2000 Proceedings, 1-7.

Silver, D., & Wang, X. (1997). Tracking and visualizing turbulent 3D features. IEEE Taransactions on Visualization and Computer Graphics, 3(2), 129-141.

Silver, D., & Wang, X. (1998). Tracking scalar features in unstructured datasets. IEEE Visualization 1998 Proceedings, 79-86.

Strang, G. (1980). Linear Algebra and its Applications. (2nd Edition ed.). New York: Academic Press.

Symon, K. R. (1971). Mechanics. (3rd Edition ed.). Menlo Park, CA.: Addison-Wesley Publishing Company.

Villasenor, J., & Vincent, A. (1992). An algorithm for space recognition and time tracking of vorticity tubes in turbulence. CVGIP: Image Understanding, 55(1), 27-35.

Youngs, D. L. (1984). Numerical simulation of turbulent mixing by Rayleigh-Taylor instability. Physica D, 12, 32-44.

Zabusky, N. J. (1999). Vortex paradigm for accelerated inhomogeneous flows: Visiometrics for the Rayleigh-Taylor and Richtmyer-Meshkov Environments. Annual Review of fluid Mechanics, 31, 495-536.